

### REMARKS

Claims 1-20 are currently pending in the present application; claims 6, 15, 18, and 20 have been objected to as dependent upon rejected based claims.

No claims have been amended in this response. Reconsideration of the claims is respectfully requested; for the convenience of the examiner, a complete, clean copy of the claims is included as an appendix to this response.

#### I. Summary of Present Invention

The present invention is a method for deferred deletion of entries in a directory service backing store. Although shown as a preferred embodiment within the specification, the invention is not limited to a Lightweight Directory Access Protocol (LDAP) directory service provided with a DB/2 backing store. As stated in the specification, the principles of the present invention may be practiced in other types of directory services, e.g., X.500, and using other relational database management systems, e.g., Oracle, Sybase, Informix, etc., as the backing store.

In either the present invention or the prior art, an entry in an LDAP directory is deleted using an SQL statement. In the prior art, the directory server responds to the delete entry statement by instituting a global lock on the database tables to ensure that data in those tables cannot be modified while the entry is being deleted from the directory. In contrast, the present invention provides an enhanced delete operation whereby the entry is marked for deletion, and the actual deletion is completed at a later time.

More specifically, the present invention is a method for deleting an entry from a directory in which directory information is stored in a set of database tables; the

deletion is initiated in response to a request to delete a directory entry. In response, the directory entry is tagged in some manner as being a deleted entry, preferably by setting the entry's creation time to a null value. If a search query is received thereafter, the method excludes tagged entries from search results that would otherwise satisfy the search query. At a periodic interval, the routine then searches for tagged entries, and references to the tagged entries are then deleted throughout the set of database tables. In this manner, the completion of the entry deletion operation is deferred to enable directory queries to be processed even if deleted entries have not yet been fully expunged from the directory.

## II. 35 U.S.C. § 102(e)-Anticipation-Kennedy

The Office action has rejected claims 1-5, 7-14, 16, 17, and 19 under 35 U.S.C. § 102(e) as being anticipated by Kennedy, "System and method for managing electronic mail messages using a client-based database", U.S. Patent No. 6,134,582, issued 10/17/00 (hereinafter Kennedy). This rejection is respectfully traversed.

### Review of the teachings of the Kennedy reference

As explained in the "Summary of the Invention" section of the reference, the system that is disclosed in Kennedy provides a client-server method for managing electronic mail messages that are stored within a local, client-side database and a remote, server-side database. During a client-server session, message-related information is retrieved from the server and stored in the client-based database along with downloaded messages. Indications are provided in the client-side database as to whether a message has been

previously downloaded from the server and whether a copy of a message has been left on the server. Expiration times can be set so that messages are deleted over time. In essence, the system allows some of the information to be stored at the client and/or the server, and the system acts to synchronize the information in the local database and the remote database. Additional detail about the system in Kennedy is provided below within several portions of the patent reference that have been recited herein in support of the arguments against the claim rejections.

#### Review of the rejection of claim 1

Independent claim 1 comprises three primary elements; the claim reads as follows:

1. A method for deleting entries from a directory in which directory information is stored in a set of database tables, comprising the steps of:

responsive to a request to delete a directory entry, tagging the directory entry in a first table;

periodically searching for tagged entries in the first table during a cleanup process interval; and

deleting references to the tagged entries throughout the set of database tables.

Against the first element of claim 1, the rejection cites the following two sections of Kennedy in support of an anticipation argument that Kennedy discloses the element: column 11, line 48, to column 12, line 8; and column 13, line 33, to column 14, line 68. In addition, the rejection states: "(According to the specification, the directory service is a 'White pages' lookup provided by some e-mail clients)". This statement at the beginning of the rejection of claim 1 appears

to be the foundation for an anticipation argument against all of the rejected claims based on Kennedy, not just the first element of independent claim 1. However, Applicant strongly disagrees with this premise and argues that because this premise is incorrect, then the basis for rejecting any claim based on Kennedy is improper in view of this error alone, although Applicant provides additional arguments against the rejection further below.

It is important to note that all of the claims in the present application are directed to a directory service or a method or a computer program product that involves a directory, but Kennedy does not mention a directory or a directory service even once, even though it has been used as an anticipatory reference. The rejection appears to recognize this fact, and in an attempt to remedy this deficiency in Kennedy, the rejection includes the above-noted statement: "(According to the specification, the directory service is a 'White pages' lookup provided by some e-mail clients)".

In other words, it appears as if this statement is an attempt to argue that Kennedy inherently or implicitly discloses essential elements of the claimed invention concerning a directory or a directory service, even though Kennedy does not explicitly disclose these essential elements. The implicit logic of the rejection appears to be the following:

(A) Kennedy does not explicitly disclose a directory or a directory service;

(B) Kennedy does disclose an e-mail client;

(C) All e-mail clients include a directory or a directory service;

(D) Therefore, Kennedy inherently discloses a directory or a directory service.

It should be readily apparent that the logic of the rejection is defective. It is true that the specification of the present application states on page 1: "Still another directory service is a 'white pages' lookup provided by some e-mail clients, e.g., Netscape Communicator, Lotus Notes, Endora and the like)." This statement merely presents the fact that a directory service can be used as a "white pages lookup", and this statement then proceeds to provide some examples of software applications that act as e-mail clients.

However, not all e-mail clients include the feature of a "white pages lookup". In other words, the fact that a system includes an e-mail client does not necessarily imply that the system includes the feature of a "white pages lookup". In addition, although many white-page-lookup functions might be implemented using a directory service, not all white-page-lookup functions are implemented using a directory service. In other words, the fact that a system includes a white-page-lookup function does not necessarily imply that the system includes the feature of a directory service.

Therefore, although Kennedy may disclose an e-mail client, it is not necessarily the case that the system disclosed in Kennedy inherently includes a directory service. Hence, the argument that Kennedy discloses the first element of independent claim 1 is deficient. Kennedy does not disclose a directory or a directory service, so it is not possible for Kennedy to disclose the elements "responsive to a request to delete a directory entry" or "tagging the directory entry in a first table" as recited in the first element of independent claim 1.

Moreover, a directory or directory service is mentioned in multiple elements throughout all of the claims of the present application. Hence, the basis for the anticipation

rejection of all of the rejected claims is deficient because the applied reference does not disclose multiple elements of the claims.

As stated at MPEP § 2131: "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Because Kennedy fails to disclose, either explicitly or implicitly, multiple elements of the claimed invention, the rejection is erroneous, and Applicant requests the withdrawal of the rejection.

Dependent claims 2-5, 7, and 8 are dependent upon independent claim 1 and include the elements of claim 1. Hence, because Kennedy does not disclose all of the features of independent claim 1 as required by a proper anticipatory reference, Kennedy is also deficient as an anticipatory reference against claims 2-5, 7, and 8, and Applicant also requests the withdrawal of the rejection of these claims (dependent claim 6 also depends from claim 1, but claim 6 has not been rejected, merely objected, so claim 6 is temporarily ignored here in this argument).

Applicant also provides additional arguments concerning other deficiencies in the rejection of other claims besides independent claim 1. The following arguments essentially show that the portions of Kennedy that have been cited against specific claim elements do not disclose the claim elements, and, therefore, Kennedy is deficient as an anticipatory reference for additional reasons.

With respect to dependent claim 2, the rejection cites two sections (column 11, line 48, to column 12, line 8; and column 13, line 33, to column 14, line 68) of Kennedy in support of an anticipation argument that Kennedy discloses claim 2, which reads: "wherein the directory entry is tagged by setting its creation time to a given value". Applicant notes that, in almost two columns of patent text, there are no features of the disclosed system that are even slightly similar to the claimed feature. The cited portions of Kennedy mention that a date and time field is compared against a preset period of time, and based on the results of the comparison, a "delete" flag might be set. However, as should be apparent by a simple reading of the passage in Kennedy, the delete flag is separate from the date and time field; the date and time field is not used to indicate that the entry should be deleted.

With respect to dependent claim 3, the rejection cites the same two sections (column 11, line 48, to column 12, line 8; and column 13, line 33, to column 14, line 68) of Kennedy in support of an anticipation argument that Kennedy discloses the claim, which reads: "wherein the given value is a null value". Claim 3 is dependent on claim 2; Kennedy does not disclose the use of a creation time as a delete tag, as required by claim 2, and Kennedy does not disclose that the creation time is set to a null value, as required by claim 3.

Against the first element of dependent claim 4, which comprises the element of "performing a search for directory entries that satisfy a search query", the rejection again cites a portion of Kennedy at column 13, line 66, to column 14, line 47; as noted above, Kennedy does not disclose the use of directory entries. Against the second element of dependent claim 4, which comprises the element of "excluding tagged

entries from search results that otherwise satisfy the search query", the rejection cites a portion of Kennedy at column 21, line 54, to column 23, line 18. The basis of argument for this rejection is entirely unclear because the rejection cites approximately one and a half columns of patent text that discusses a method of assembling a message that comprises multiple message parts; nothing in the cited portion even remotely approaches a feature that is similar to the claimed element. Applicant also notes that the cited portion appears to be randomly selected and randomly applied against the claim element because the cited portion includes the concluding paragraphs of the patent and the first several lines of the first claim.

Against dependent claim 5, which states that "the step of excluding tagged entries includes modifying an SQL query to exclude rows having a null change creation", the rejection cites a portion of Kennedy at column 8, line 51, to column 9, line 63, which is provided immediately below (Applicant herein recites the entire section of more than a column of patent text to emphasis what appears to be a random selection of a portion of the reference because it has no relevance to the claimed feature):

In FIG. 3, a remote computer 49 operates as a server and generally includes an e-mail server application 110, a local store 115, and a client manager control 120. In an exemplary embodiment, the server 49 is a POP3 mail server, but it will be appreciated that the present invention is not limited to this type server. In the exemplary embodiment, the client 20 includes a local message store 38, a database 39, an e-mail program module 36, and a message manager program module 37 for facilitating message management and operation of the database 39.

With respect to the exemplary embodiment, the client 20 provides two modes of operation in connection with the server 49. These modes are a default mode and a "leave on server" mode. In the default mode, the client 20



sends a delete command to the server 49 to delete a message from the server 49 after the message has been downloaded to the client 20. In the "leave on server" mode, the client does not send a delete command to the server 49 after the message has been downloaded to the client 20, thereby allowing the message to remain on the server 49 although the message has been downloaded. The mode of operation generally is selected based on user-preference. Advantageously, the present invention optimizes the management of messages when the client 20 is in the "leave on server" mode, as will be described below in connection with FIGS. 4-8.

The server 49 houses any e-mail messages from clients in the local store 115 while awaiting transmission to an appropriate destination. The e-mail server application 110 forwards messages over the WAN 52 from a sender client (not shown) to the client 20, upon request by the client 20. The client manager control 120 is a program used to set up computer systems, such as clients 1, 11a, 11b, 11c (FIG. 1), and 20 (FIG. 2) on the network. The client manager control 120 can also specify the addresses of the computer systems located on the network. In addition, the client manager control 120 typically facilitates the management of incoming and outgoing messages on the server. When a request for a message is made by the client 20 to the server 49, the e-mail server application 110 on the server 49 responds by retrieving the message from the local store 115 on the server 49 and by transmitting the message over the WAN 52 to the client 20. The message is then downloaded into the local message store 38 located at the client 20. The local message store 38 houses all downloaded messages from the server 49.

During the download operation, data fields are populated within the database 39 with message-related information associated with the downloaded message. The information includes a unique identifier for identifying the message, a session identifier for indicating the particular order in which the message is retrieved from the server, a message size and other message-related information that will be described in greater detail herein below with respect to FIGS. 4-8. The e-mail program module 36 provides facilities for creating, addressing, sending, receiving, and forwarding messages, while the message manager program module 37 manages messages during download and deletion operations utilizing the database 39. The use of the database 39 is

described in greater detail in connection with FIGS. 4a-4k, collectively described as FIG. 4.

With continuing reference to FIGS. 1-3 and now turning to FIGS. 4a-4k, a client-based database used in connection with the exemplary program module 37 is illustrated. FIGS. 4a-4k illustrate a client-based database for archiving messages in accordance with an exemplary embodiment of the present invention.

The database 39 can include multiple data fields, organized within an array structure, for maintaining message-related information. To support download and delete operations, typical data fields of the database include: a session identifier (session ID) 200, a unique identifier (UIDL) 205, a message size 210, an entry identifier (EID) 215, a receive date and time 220, which is the local machine time, an "on server" flag 225, a "download" flag 230, and a "delete" flag 235. Message re-assembly operations can be supported by adding to the database structure certain data fields corresponding to portions of a MIME-compatible message, such as a message group identifier (message group ID) 240, a message part number 245, and a total parts number 250.

It is entirely unclear how this passage discloses anything remotely related to the modification of an SQL query.

With respect to independent claim 9, this claim is similar to independent claim 1 and dependent claim 4 in that claim 9 is a combination of the elements of claims 1 and 4. The rejection merely cites the same portions of Kennedy against claim 9 that were cited against claims 1 and 4. Applicant has refuted these arguments above with respect to claims 1 and 4, and Applicant maintains that independent claim 9 is not anticipated by Kennedy for the same reasons.

With respect to independent claim 14, this claim is similar to independent claim 1 and dependent claim 4 in that claim 14 is a subset of the combination of the elements of claims 1 and 4. The rejection merely cites the same portions of Kennedy against claim 14 that were cited against claims 1 and 4. Applicant has refuted these arguments above with respect to claims 1 and 4, and Applicant maintains that

independent claim 14 is not anticipated by Kennedy for the same reasons.

With respect to independent claim 16, this claim is similar to independent claim 1 in that independent claim 1 is a method claim, whereas independent claim 16 is directed to a computer program product comprising means for performing the steps of a process similar to that claimed in claim 1. Independent claim 16 has been rejected using the same argument as independent claim 1, and Applicant maintains that independent claim 16 is not anticipated by Kennedy for the same reasons as independent claim 1 that were provided above.

With respect to independent claim 19, this claim is similar to independent claim 1 and dependent claim 4 in that claim 19 is a subset of a combination of the elements of claims 1 and 4 and in that independent claim 1 is a method claim while independent claim 19 is directed to a directory service comprising means for performing the steps of a process similar to that claimed in claims 1 and 4. The rejection merely cites the same portions of Kennedy against claim 19 that were cited against claims 1 and 4. Applicant has refuted these arguments above with respect to claims 1 and 4, and Applicant maintains that independent claim 19 is not anticipated by Kennedy for the same reasons.

Dependent claims 10-13 and 17 were rejected using the same arguments as dependent claims 2-4, 7, and 8. Applicant maintains that dependent claims 10-13 and 17 are not anticipated by Kennedy for the same reasons as dependent claims 2-4, 7, and 8 that were provided above.

Clearly, the rejection has not carefully considered the elements of the claims nor has the rejection pointed out the claimed features within Kennedy as is required for a proper

anticipation rejection. Applicant requests the withdrawal of the rejection against all rejected claims.

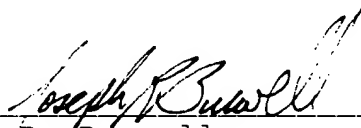
### III. Conclusion

It is respectfully urged that the present patent application is patentable, and Applicant kindly requests a Notice of Allowance.

For any other outstanding matters or issues, the examiner is urged to call or fax the below-listed telephone numbers to expedite the prosecution and examination of this application.

DATE: February 25, 2002

Respectfully submitted,



Joseph R. Burwell

Reg. No. 44,468

ATTORNEY FOR APPLICANT

Law Office of Joseph R. Burwell

P.O. Box 28022

Austin, Texas 78755

(512) 502-9448 (voice)

(512) 597-1218 (fax)

## APPENDIX

1. A method for deleting entries from a directory in which directory information is stored in a set of database tables, comprising the steps of:

responsive to a request to delete a directory entry, tagging the directory entry in a first table;

periodically searching for tagged entries in the first table during a cleanup process interval; and

deleting references to the tagged entries throughout the set of database tables.

2. The method as described in claim 1 wherein the directory entry is tagged by setting its creation time to a given value.

3. The method as described in claim 2 wherein the given value is a null value.

4. The method as described in claim 1, further including the steps of:

performing a search for directory entries that satisfy a search query; and

excluding tagged entries from search results that otherwise satisfy the search query.

5. The method as described in claim 4 wherein the step of excluding tagged entries includes modifying an SQL query to exclude rows having a null change creation.

6. The method as described in claim 1 wherein the directory is a Lightweight Directory Access Protocol (LDAP) directory service and the database tables are managed by a relational database management service.

7. The method as described in claim 1 wherein the first table is an entry table.

8. The method as described in claim 7 wherein the set of database tables includes at least one attribute table storing information about an attribute.

9. A method for deleting entries from a directory in which directory information is stored in a set of database tables, comprising the steps of:

responsive to a request to delete a directory entry,  
tagging the directory entry in a first table;

responsive to a search for directory entries that satisfy a search query, excluding tagged entries from search results that otherwise satisfy the search query;

periodically searching for tagged entries during a cleanup process interval; and

deleting references to the tagged entries throughout the set of database tables.

10. The method as described in claim 9 wherein the directory entry is tagged by setting its creation time to a given value.

11. The method as described in claim 10 wherein the given value is a null value.

12. The method as described in claim 9 wherein the first table is an entry table.

13. The method as described in claim 12 wherein the set of database tables includes at least one attribute table storing information about an attribute.

14. A method for searching a database from a directory service, comprising the steps of:

responsive to a search for directory entries that satisfy a search query, excluding given entries from search results that otherwise satisfy the search query, wherein the given entries identify database entries that have been tagged for deletion; and

returning the search results.

15. The method as described in claim 14 where in the directory service is a Lightweight Directory Access Protocol (LDAP) directory service and the database tables are managed by a relational database management service.



16. A computer program product in a computer-readable medium for deleting entries from a directory in which directory information is stored in a set of database tables, comprising:

means responsive to a request to delete a directory entry for tagging the directory entry in a first table;

means for periodically searching for tagged entries in the first table during a cleanup process interval; and

means for deleting references to the tagged entries throughout the set of database tables.

17. The computer program product as described in claim 16, further including:

means responsive to a search for directory entries that satisfy a search query for excluding tagged entries from search results that otherwise satisfy the search query.

18. The computer program product as described in claim 17 wherein the search query is a Lightweight Directory Access Protocol (LDAP) directory service query.

19. A directory service, comprising:

a directory organized as a naming hierarchy having a plurality of entries each represented by a unique identifier;

a relational database management system having a backing store for storing directory data in a set of database entries; and

means for deleting entries from the directory, comprising:

means responsive to a request to delete a directory entry for tagging the directory entry in a first table;

means for periodically searching for tagged entries in the first table during a cleanup process interval;

means for deleting references to the tagged entries throughout the set of database tables; and

means responsive to a search for directory entries that satisfy a search query for excluding tagged entries from search results that otherwise satisfy the search query.

20. The directory service as described in claim 19 wherein the directory is compliant with the Lightweight Directory Access Protocol (LDAP).